Automatic Control II

Computer exercise 2

Stationary stochastic processes
and state estimation

Preparations:
- Read Chapter 5, and in particular Sec. 5.8, in the textbook by Glad and Ljung.
- Preparation exercises 1–4 in Section 3.
1 Introduction

Many control systems are subject to random disturbances. To model and understand their effects on a dynamics system, the theory of stochastic processes (also called random processes) is very useful. In this computer laboratory properties of (mainly) discrete-time stationary stochastic processes will therefore be studied, in order to give a feasible background for understand the effects of random disturbances.

2 Background: Basic concepts

2.1 Deterministic vs stochastic signals

The most characteristic feature of a disturbance is that its value is not known beforehand. Hence a deterministic model like, for example, \( w(t) = \sin(t) \) is seldom a good way of describing a disturbance. Instead, it is more natural to use stochastic (or random) concepts to describe disturbances.

As a simple example of a stochastic process consider the output signal generated by tossing a coin 5000 times (+1 for heads, -1 for tails). We will obtain a different output sequence every time we do the experiment. The output from each experiment is called a realization of the stochastic process.

As a stochastic process represents a whole family of signal realizations, the deterministic signal descriptions can not be directly applied on a stochastic process. The search for good ways of characterizing a stochastic process has been a long and arduous one, but in the last hundred years the mathematicians and statisticians have been able to develop a whole framework for stochastic processes that allows us to describe them in a way that resembles that of deterministic signals in many ways. The whole framework is built around the so called covariance function.

For a stochastic signal\(^1\) \( w(t) \) the function

\[
m_w = \mathbb{E} w(t)
\]

is called the mean value function. The (auto) covariance function is defined as

\[
r_w(\tau) = \text{cov}(w(t+\tau), w(t)) = \mathbb{E} [(w(t+\tau) - m_w)(w(t) - m_w)]
\]

Note that \( r_w(0) \) is the variance of the process and tells how large the fluctuations are (the standard deviation is the square root of \( r_w(0) \)). It follows from Schwartz’s inequality that \( |r_w(\tau)| \leq r_w(0) \). The value of \( r_w(\tau) \) gives the correlation between values of the process with a time spacing of \( \tau \). Values close to \( r_w(0) \) mean a strong correlation, zero values indicate no correlation and negative values indicate negative correlation.

\(^1\)We will only consider stochastic processes that is wide sense stationarity, then both the mean and covariance functions are independent of time.
2.2 Characterizing the coin tossing stochastic process

The coin tossing process is a typical stochastic processes, so let us take a closer look at it. We know that we get different realizations every time we run the process, but what is it that characterizes all the realizations? The key characteristic is that each toss is independent from the others, i.e., there is no correlation between one toss and another. In mathematical terms, we can describe this process as a sequence of independent identically distributed (iid) random variables, with mean value $m = 0$, and variance $\sigma^2 = 1$. The covariance function hence becomes

$$r_w(\tau) = E[(y(t + \tau) - m)(y(t) - m)] = \begin{cases} 0 & \text{for } \tau \neq 0 \\ \sigma^2 & \text{for } \tau = 0 \end{cases}$$

The correlation function for the coin tossing process simply says that the coin tossing is uncorrelated (i.e., the chance of getting a head or tail does not depend on what you got in the previous tossing) and has a variance of $\sigma^2 = 1$.

2.3 Spectral density of stochastic process

In the time domain, a stochastic process $w(t)$ is normally characterized by its mean $m$ and covariance function $r_w(\tau)$. If we take the Fourier transform of the covariance function, we obtain the spectra of the process, which describes the frequency content of the stochastic process in a similar way that the Fourier transform does for deterministic signals. The spectral density of a stochastic process with covariance function $r_w(\tau)$ is defined as

$$\phi_w(\omega) = \sum_{\tau = -\infty}^{\infty} r_w(\tau)e^{-i\omega\tau}$$

The covariance function can be found from the spectral density by the inverse relation

$$r_w(\tau) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi_w(\omega)e^{i\omega\tau} d\omega$$

The area under the spectral density curve represents the mean signal power in a certain frequency band. In particular, we have that the variance of the signal is given by

$$r_w(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi_w(\omega)d\omega$$

The coin tossing process has a covariance function that is a unit impulse sequence. The Fourier transform of that is a constant, i.e., it has equal amounts of all frequency components.

The coin tossing process is an example of a white noise process, i.e., a sequence of independent random variables with a certain probability distribution. However, as a disturbance model, the coin tossing is not very realistic (a random sequence with -1 and 1). A more suitable pattern is obtained if the disturbance is modeled...
as a white Gaussian process with zero mean and variance $\sigma^2$. Such a process is often denoted

$$w(t) \sim N(0, \sigma^2)$$

### 2.4 Filtering of stochastic processes

When using stochastic processes for disturbance modelling it is of utmost importance to know and understand how the characteristics of a stationary stochastic process change as the process is filtered by a linear system, with transfer function

$$H(q) = \frac{B(q)}{A(q)} = \sum_{n=0}^{\infty} h(n)q^{-n}.$$  \hspace{1cm} (7)

In particular, we need to know how the mean, covariance function and spectrum change. The main results are summarized in Figure 1. Notice, though, that in practice it is seldom a good way to compute auto/cross covariance functions using the formulas in Figure 1.

\[ r_{yu}(\tau) = \text{cov}(y(t + \tau), u(t)) = \sum_{n=0}^{\infty} h(n)r_u(\tau - n) \]

\[ r_u(\tau) = \text{cov}(u(t + \tau), u(t)) \]

\[ r_y(\tau) = \text{cov}(y(t + \tau), y(t)) = \sum_{n=0}^{\infty} \sum_{l=0}^{\infty} h(n)h(l)r_u(\tau + l - n) \]

\[ \phi_{yu}(\omega) = \sum_{\tau=-\infty}^{\infty} r_{yu}(\tau)e^{-i\tau\omega} = H(e^{i\omega})\phi_u(\omega) \]

\[ \phi_u(\omega) = \sum_{\tau=-\infty}^{\infty} r_u(\tau)e^{-i\tau\omega} \]

\[ \phi_y(\omega) = \sum_{\tau=-\infty}^{\infty} r_y(\tau)e^{-i\tau\omega} = H(e^{i\omega})H(e^{-i\omega})\phi_u(\omega) \]

**Figure 1**: Summary of filtering results for scalar stochastic processes.

### 2.5 Disturbance models

#### 2.5.1 Transfer functions and ARMA models

A large class of disturbances can be described by filtering a white noise process. This is obtained by letting $u(t)$ in Figure 1 be white noise. Assume that $A(q)$
which shows how average: the dependence on

This is an ARMA process/model. ARMA stands for auto-regressive moving

Two special cases of ARMA processes are the AR process and the MA process:

In accordance with the results in Figure 1, the spectrum of \( y(t) \) is

\[
\phi_y(\omega) = H(e^{i\omega})H(e^{-i\omega})\phi_u(\omega) = \frac{B(e^{i\omega})B(e^{-i\omega})}{A(e^{i\omega})A(e^{-i\omega})}R_u,
\]

where \( R_u = \phi_u(\omega) \) is the variance = constant spectrum of the white noise input \( u(t) \).

2.5.2 State space models

A stochastic process which can be represented as \( y(t) = H(q)u(t) \), where \( u(t) \) is white noise, and \( H(q) = \frac{B(q)}{A(q)} \) is a rational function in \( q \), can also be represented by a state space model,

\[
x(t+1) = Fx(t) + Gu(t), \\
y(t) = Hx(t) + Ju(t).
\]

Here \( F, G, H, J \) are any matrices such that \( H(q) = \frac{B(q)}{A(q)} = H(qI - F)^{-1}G + J \).

A typical situation considered in this course is where a system depends on an external input, \( u(t) \), as well as a process disturbance. When using feedback for control the closed loop system will also be affected by measurement disturbances. Thus, there are in general two disturbance sources, entering the system in two different ways. Both of these may be modeled as filtered white noise,
as described above. Combining these disturbance models with the model representing the input–output dynamics, a total model of the overall behavior of the system is obtained.

It is convenient to use state space representations for such a total model, and in particular the “standard model”,

\[ x(t + 1) = Fx(t) + Gu(t) + Nv_1(t), \]
\[ y(t) = Hx(t) + v_2(t), \]  \hspace{1cm} (9)

where \( v_1(t) \) and \( v_2(t) \) are zero mean white noise processes with covariances

\[ E[v_1(t)v_1^T(t)] = R_1, \quad E[v_2(t)v_2^T(t)] = R_2, \]

and cross-covariance \( E[v_1(t)v_2^T(t)] = R_{12} \). In this computer exercise we will disregard from the input \( u(t) \) for brevity (assume that \( u(t) = 0 \)), so the state space model to be considered is

\[ x(t + 1) = Fx(t) + Nv_1(t), \]
\[ y(t) = Hx(t) + v_2(t). \]  \hspace{1cm} (10)

In order to analyze the statistical properties of this model it is handy to use the covariance matrix and the covariance function of the state vector \( x(t) \). The covariance matrix, \( \Pi_x = E[x(t)x^T(t)] \), is readily obtained as the solution of the discrete-time Lyapunov equation:

\[ \Pi_x = F\Pi_x F^T + NR_1N^T. \]  \hspace{1cm} (11)

The covariance function then is

\[ r_x(\tau) = F^T\Pi_x, \quad \text{for} \quad \tau \geq 0. \]  \hspace{1cm} (12)

For \( \tau < 0 \) one can use that \( r_x(-\tau) = r_x(\tau)^T \).

Also continuous-time disturbances can be regarded (and modeled) as stochastic processes. Although the theory of continuous-time stochastic processes is much more intricate than for discrete-time stochastic processes, the results are very similar to the latter case, and are straightforward to use. In particular, the “standard model” is the state space representation

\[ \dot{x}(t) = Ax(t) + Bu(t) + Nv_1(t), \]
\[ y(t) = Cx(t) + v_2(t), \]  \hspace{1cm} (13)

and the covariance matrix of the state vector, \( \Pi_x = E[x(t)x^T(t)] \), is (assume that \( u(t) = 0 \)) the solution of the continuous-time Lyapunov equation,

\[ 0 = A\Pi_x + \Pi_x A^T + NR_1N^T. \]  \hspace{1cm} (14)
3 Preparation exercises

Preparation exercise 1.

(a) Determine a first order filter, \( H(q) = \frac{b}{q+a} \), such that the stochastic process

\[
y(t) = H(q)u(t), \quad E u(t) = 0, \quad r_u(\tau) = \begin{cases} 
1 & \text{for } \tau = 0, \\
0 & \text{for } \tau \neq 0,
\end{cases}
\]

has the spectral density

\[
\phi_y(\omega) = \frac{0.75}{1.25 - \cos \omega}.
\]

(b) What is the variance of the signal \( y(t) \), i.e. what is \( r_y(0) \)? In Exercise 1, you will need the results from here.

*Hint:* In order to find the variance, write down a state space realization of (15) and solve the Lyapunov equation (11).

**Answer:**
**Preparation exercise 2.** Determine the covariance function \( r_y(\tau) \) for the AR process

\[
y(t + 1) + ay(t) = e(t), \quad \Leftrightarrow \quad \begin{cases} 
x(t + 1) = -ax(t) + e(t), \\
y(t) = x(t),
\end{cases}
\]

where \( e(t) \) is white noise with zero mean and unit variance.

**Answer:**

**Preparation exercise 3.**

(a) Use the help function and/or read the reference page for the Matlab function **dlyap** to find out how it can be used for computation of the covariance matrix of the state vector for the discrete-time stochastic process (10).

(b) Use the help function and/or read the reference page for the Matlab function **lyap** to find out how it can be used for computation of the covariance matrix of the state vector for the continuous-time stochastic process (13).

**Notes:**
Preparation exercise 4. Assume that an observer should be used for state estimation for the discrete-time stochastic process (10), or the continuous-time stochastic process (13).

(a) Use the help functions and/or read the reference pages for the Matlab functions acker and place to find out how they can be used for computation of the observer gains $K$, given some pre-defined desired observer poles.

(b) Use the help function and/or read the reference page for the Matlab function dare to find out how it can be used for computation of the Kalman gain $K$ for the Kalman filter (predictor) for (10).

(c) Use the help function and/or read the reference page for the Matlab function care to find out how it can be used for computation of the Kalman gain $K$ for the Kalman filter for (13).

Notes:

\[ H_1(q) = \frac{b}{q + a}, \quad H_2(q) = \frac{b_0q + b_1}{q^2 + a_1q + a_2} \quad (16) \]

4 Computer exercises

4.1 Filtering a white stochastic process

In this section we will analyze the covariance function, the spectrum and different realizations of a stochastic process that is obtained by filtering a white stochastic process with the first and second order systems below.

In particular, we will look at how system parameters such as poles and zeros influence the results.
Exercise 1.

(a) Run the macro `lab2a(a,b)` and check if the filter you calculate in Preparation exercise 1 is correct.

(b) Run the MATLAB macro `noise` and answer the following questions onward. Note that $b \equiv \sqrt{3}/2$ is a fixed value. Vary the pole of $H_1(q)$ on the real axis. What happens when the pole is close to $+1$, positive but close to the origin, and when it is negative? What happens when the pole is in the origin? Try to explain that.

(c) Where should you place the poles of $H_1(q)$ and $H_2(q)$ to get a low-pass filter?

(d) Where should you place the poles of $H_2(q)$ to get a resonance peak at $\omega = 1$? What can you say about the frequency content of the signal by just looking at the realization?

(e) How does a resonant system manifest its properties in the different diagrams?

(f) What happens when $H_2(q)$ has a zero close to the unit circle?

Answer:

4.2 Calculating and estimating covariance functions

The relations in Figure 1, and Equations (11)–(12), give expressions for how the theoretical statistical properties for a (discrete-time) stochastic process can be computed. When dealing with experimental data, or data from simulations, it is often useful to compute *empirical* or *estimated* values of statistical measures.
Say that \( y(1), \ldots, y(N) \) are data obtained from a simulation or experimental measurements. You are probably familiar with the expression

\[
\hat{m}_y = \frac{1}{N} \sum_{k=1}^{N} y(k)
\]

for computation of the empirical (estimated) mean value of \( y(t) \). The covariance function \( r_y(\tau) \) can be estimated in a similar way:

\[
\hat{r}_y(\tau) = \frac{1}{N-\tau} \sum_{k=1}^{N-\tau} (y(k + \tau) - \hat{m}_y)(y(k) - \hat{m}_y).
\]

In simulations realizations of stochastic processes are typically generated as filtered white noise, i.e. the output of a linear filter with white noise as input. In Matlab linear filters can be represented eg. as LTI objects, and the filtering can then be performed with the function `lsim`.\(^2\) The white noise input is easily generated by the Matlab function `randn`, which generates a vector/matrix whose entries are independent (pseudo) random numbers, with Gaussian distribution, zero mean and unit variance.

**Exercise 2.**

(a) Generate a white noise vector by typing

\[
y = 2 \times \text{randn}(1,1000);
\]

You should now have a row vector with 1000 elements, which represents a white noise sequence. What is the variance of the white noise? Use the Matlab function `cov` to compute the variance of \( y \).

(b) Suppose that \( a \) has a value, and that a white noise vector is generated by

\[
y = a \times \text{randn}(1,1000);
\]

What value should \( a \) have in order to get a white noise sequence of variance equal to 10?

**Answer:**

---

\(^2\)Another useful Matlab function is `filter`. 

10
Exercise 3. With the function `emp_covfunc` you can simulate, and estimate the covariance function of the first order stochastic (AR) process

\[ y(t+1) + ay(t) = e(t) \iff y(t) = \frac{1}{q-a}e(t), \quad (17) \]

where \( e(t) \) is zero mean, unit variance white noise. The function has the following syntax:

\[
\text{emp_covfunc}(a,N,nr,tau);
\]

Type `help emp_covfunc` for details. Default values are \( N=100, nr=1 \) and \( tau=50 \). For example, to generate a process with a pole in 0.9 use the following syntax:

\[
\text{emp_covfunc}(-0.9);
\]

Inspect how the quality of the estimated covariance function vary with the number of data \( N \) (try \( N = 100, 1000, 10^4, \ldots \)) and the time shift \( \tau \) for different pole locations. Especially, verify that the estimated covariance functions tends to the true ones, as the number of data tends to infinity. Also check if the results you obtained in Preparation exercise 2 are correct. Repeat the estimation procedure to study the effect of the individual realizations.

Answer:
4.3 State estimation

4.3.1 Observers

A discrete-time system has the state space representation

\[ x(t+1) = Fx(t) + Nv_1(t), \]
\[ y(t) = Hx(t) + v_2(t), \]  

(18)

with

\[ F = \begin{bmatrix} 0.3 & -0.5 \\ 1 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \]

The process noise, \( v_1 \), and the measurement noise, \( v_2 \), are uncorrelated, zero mean and unit variance white noise processes. Hence,

\[ R_1 = 1, \quad R_2 = 1, \quad R_{12} = 0. \]

The standard observer for state estimation is

\[ \dot{x}(t+1) = F\dot{x}(t) + K(y(t) - H\hat{x}(t)). \]  

(19)

Exercise 4. Create the LTI object \( \text{sys} \) as a representation of the system (18) by typing

\[ \text{sys}=\text{ss}(F,N,H,0,-1); \]

Of course you need to define the system matrices \( F, N \) and \( H \) first. (The -1 in the last argument defines the sampling interval as unspecified.)

Next, determine an observer gain \( K \) so that the observer poles are placed in \( 0.1 \pm i0.45 \). Use eg. \texttt{place} or \texttt{acker}. (See Prep. exerc. 4 (a))

Answer:
The state estimation error $\tilde{x} = x - \hat{x}$ evolves in time as

$$\tilde{x}(t+1) = (F-KH)\tilde{x}(t) + Nv_1(t) - Kv_2(t) = (F-KH)\tilde{x}(t) + \begin{bmatrix} N & -K \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix}.$$  

The size of $\tilde{x}$ is one measure of the quality of the estimator/observer.

Another interesting signal is the output innovations, which for the observer (19) is defined as

$$\nu(t) = y(t) - H\hat{x}(t) = H\tilde{x}(t) + v_2(t).$$  

The output innovations represent the new information that is fed into the observer — the observer (19) can be written as

$$\hat{x}(t+1) = F\hat{x}(t) + K\nu(t).$$

**Exercise 5.**

(a) Use the function `dlyap` to compute the covariance matrix of the state estimation error, $\Pi_{\tilde{x}} = E\tilde{x}\tilde{x}^T$, for the observer (19), with the observer gain $K$ you got in Exercise 4.

*Hint:* Use the matrices in Eq. (20). Also see, Prep. exerc. 3.

(b) Use the function `innov_char` to investigate the statistical properties of the output innovations, $\nu$, for the observer. Use the syntax

```
innov_char(sys,K,R1,R2);
```

The `innov_char` function can be used with output arguments also (use `help` to see how), but without output arguments the function will produce plots with the covariance function and the spectrum of the output innovations, $\nu$. What does these look like?

**Answer:**

13
4.3.2 Optimal state estimation: The Kalman filter

The optimal observer, that minimizes the covariance of the state estimation error, $E\tilde{x}\tilde{x}^T$, is the *Kalman filter*. In order to be perfectly clear of how an estimate of the state vector is obtained, the following notation is used: $\hat{x}(t|s)$ is the estimate of $x(t)$ based on the measurements ..., $y(s-1), y(s)$, i.e. measurements up to time $s$. By this notation the standard observer (19) provides $\hat{x}(t|t-1)$, or actually the prediction $\hat{x}(t+1|t)$. With a slight abuse of language we here call the optimal observer a *Kalman filter*, i.e. the observer that minimizes $E\tilde{x}(t|t-1)\tilde{x}^T(t|t-1)$.$^3$

**Exercise 6.** (a) Use the function `dare` to compute the Kalman gain $K_{KF}$ and the optimal covariance of the state estimation error, $P = E\tilde{x}(t|t-1)\tilde{x}^T(t|t-1)$.

(b) Now use `dlyap` to compute the covariance of the state estimation error for the Kalman filter. Compare the result with the $P$ you got in (a) — are they identical?

(c) Compare $P$ with $\Pi_{P}$ from Exercise 5. Does $P < \Pi_{P}$ hold?

---

$^3$Strictly speaking, a Kalman filter gives $\hat{x}(t|t)$, while $\hat{x}(t|t-1)$ is obtained by the Kalman predictor.
Exercise 7. The output innovations of a Kalman filter, $\nu(t) = y(t) - H\hat{x}(t|t-1)$, are white noise. How does that manifest itself in the covariance function and the spectrum of $\nu$? Use `innov_char` to investigate these properties for your Kalman filter.

Answer:

Exercise 8. Optional

Generate white noise sequences for $v_1$ and $v_2$ (use `randn`). Then simulate the system:

```plaintext
[z,t,x]=lsim(sys,v1);
y=z+v2;
```

Simulate and compare the pole placement observer and the Kalman filter, e.g. by plotting $x$ and $\hat{x}$ together, or simply $\tilde{x}$. (You need to create LTI objects for the observer and the Kalman filter, and/or the corresponding state equations for $\tilde{x}$, as in (20). Then use `lsim` with $y$ as input.)

Answer:
Exercise 9. Optional

Use the script `observer_script` to compare a pole placement observer and the Kalman filter for some random discrete-time systems. Open the m-file to see what the script does.

Answer: